



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/752,121	12/29/2000	Julio Estrada	LOT9-2000-0023 US1	8745

7590 08/10/2005

Stephen T. Keohane, Esq.  
Lotus Development Corporation  
55 Cambridge Parkway  
Cambridge, MA 27085

EXAMINER
----------

BLACKWELL, JAMES H

ART UNIT	PAPER NUMBER
----------	--------------

2176

DATE MAILED: 08/10/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

**Office Action Summary**

Application No.

09/752,121

Applicant(s)

ESTRADA ET AL.

Examiner

James H. Blackwell

Art Unit

2176

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 23 May 2005.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1-19 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-19 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 03 October 2001 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- |                                                                                         |                                                                             |
|-----------------------------------------------------------------------------------------|-----------------------------------------------------------------------------|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)             | 4) <input type="checkbox"/> Interview Summary (PTO-413)                     |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)    | Paper No(s)/Mail Date. _____                                                |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| Paper No(s)/Mail Date _____                                                             | 6) <input type="checkbox"/> Other: _____                                    |

### DETAILED ACTION

1. This Office Action is in response to amendment filed 05/23/2005.
2. Claims 1-19 remain pending. Claims 1, 13-16, and 19 are the independent claims.
3. Objections to the Specification have been withdrawn as necessitated by amendment.
4. The rejection under 35 U.S.C. 101 of Claims 18, and 19 has been withdrawn as necessitated by amendment.

### ***Claim Rejections - 35 USC § 103***

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

6. Claims 1-6, 11, and 13-19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Cannata et al. (hereinafter, Cannata, U.S. Patent No. 6,917,962, filed 06/09/2000) in view of Salas et al. (hereinafter Salas, U.S. Patent No. 6,233,600, filed 07/15/1997)

**In regard to independent Claim 1 (and similarly independent Claims 14-16, and 19), Cannata teaches a method for integrating forms including fields defining the schema of a collaboration space into said collaboration space in that Cannata allows a user to create a new workgroup using a creation template (form) which permits the user**

Art Unit: 2176

to define parameters of the workgroup, such as the name of the workgroup and the site to be created, the scope of the project being undertaken, the number of team members, etc. (Col. 4, lines 60-67). Thus, Cannata allows the use of a template (a type of form), to define a workgroup (made up of a set of parameters describing the space; i.e. a schema).

Cannata also teaches *preparing at a browser using a legacy editor a form containing fields defining said schema and hyperlinks that point to web pages* in that both primary user 30 and secondary user 40 can communicate with server 10 by means of in HTML compliant client supporting a graphical user interface and internet browser (Col. 3, lines 46-50).

Cannata does not explicitly teach using a legacy editor to edit the template (form) or *creating a form containing hyperlinks that point to web pages*. However, Salas teaches a page builder (editor) executing on a client computer that allows the user to modify templates that contain instructions leading to the rendering of HTML pages once the pages are uploaded to the server and processed (Col. 16, lines 57-64). Thus, Salas in effect allows a user to assist in providing information to a template (form) that eventually, when uploaded to the server, defines a web space on the server.

Salas does not explicitly teach that *the form (template) is saved to local storage at said browser*. However, assuming that one has created a new file (see Fig. 7, Col. 13, lines 14-26), and one exited the editor, as indicated by Salas above, then one of ordinary skill in the art at the time of invention would have found it obvious that the file would have been saved. Given a background daemon running locally on the client

machine is checking for the status of the file, it makes sense that it would more than likely be checking the status of a locally saved file (in other words, not accessing a server and checking file statuses remotely). Hence Salas also teaches *saving the file on a client machine*. It would have been obvious to one of ordinary skill in the art at the time of invention to combine the teachings of Cannata and Salas as both inventions relate to collaborative work environments. Adding the teaching of Salas allows for creating/updating server workspaces without the need to be directly connected to the server.

Cannata does not teach *dragging and dropping said form from local storage into an upload control panel in a user interface to said collaboration space*. However, Salas teaches that if the file was modified, then it must be uploaded to the server 14 using HTTP (step 610). The file upload may be done in the background or in the foreground. If done in the foreground the user will be blocked from further work on that file until the upload is complete. Once the upload is complete, the server 14 updates metadata stored in its database 20 that is associated with the file, for example, any edit lock set by the editing user is released (Col. 132, lines 60-67).

Also, once the user is finished editing the file, it may be uploaded to the server 14 to allow other users access to it. The user signals that the file should be transmitted to the server 14 by *dragging the file onto an eRoom* displayed by the browser (step 704). *Dropping the file into the displayed eRoom* invokes an ActiveX control or a background daemon process, which manages the upload of the file to the server 14 (Col. 13, lines 19-26). It would have been obvious to one of ordinary skill in the art at the time of

invention to combine the teachings of Cannata and Salas as both inventions relate to collaborative work environments. Adding the teaching of Salas allows for creating/updating server workspaces without the need to be directly connected to the server.

Cannata continues by teaching *parsing said form to identify said fields and incorporate them into said schema* in that once this template has been completed, the server creates a dedicated site (180) having the name chosen by primary user, the administration sub-system checks to see whether all the prospective group members identified by the primary user are listed on the existing intranet-user database (190). If a prospective group member is as existing intranet user, the server then sends details of the newly created dedicated site to that member of the group (secondary user) (200). The server automatically creates a link between each secondary user's personal workspace and the newly created dedicated site (Col. 5, lines 13-19). Though Cannata does not specifically use the word parse, it is apparent from the teaching that the template (form) is processed by the server to create the dedicated site.

**In regard to dependent Claim 2, Cannata does not specifically teach that *said hyperlinks providing access to files including text, graphics, images, sound, and video files*. However, Salas teaches that the product of the uploaded templates are eRooms, which are a set of connected HTML pages displayed to a user that displays project-related files, data, and discussion lists (and typical to web pages, hyperlinks) (Col. 5, lines 1-11). It would have been obvious to one of ordinary skill in the art at the time of invention to combine the teachings of Cannata and Salas, as both inventions relate to**

the creation of collaborative web-based environments through templates. Adding the teaching of Salas allows for creating/updating collaborative web-based workspaces without the need to be directly connected to the server.

**In regard to dependent Claim 3, Cannata** does not specifically teach that *said web pages residing on servers remote from the server for said collaboration space*. However, Salas teaches that the product of the uploaded templates are eRooms, which are a set of connected HTML pages displayed to a user that displays project-related files, data, and discussion lists (and typical to web pages, hyperlinks) (Col. 5, lines 1-11). It would have been obvious to one of ordinary skill in the art at the time of invention to combine the teachings of Cannata and Salas, as both inventions relate to the creation of collaborative web-based environments through templates. Adding the teaching of Salas allows for creating/updating collaborative web-based workspaces without the need to be directly connected to the server.

**In regard to dependent Claim 4, neither Cannata nor Salas** specifically teach that *responsive to user selection of a said hyperlink, downloading linked code; and rendering said linked code into a display at said browser*. However, it is notoriously well known that selecting a hyperlink from a browser usually begins the process of downloading the page and its contents and eventually produces a rendering of the page on the browser.

**In regard to dependent Claim 5, Cannata** fails to explicitly teach that *said collaboration space being implemented within an object model selectively including place, room, folder, page, member, form, field, placetype, roomtype, skin, and placebot*

*objects*. However, Salas teaches a collaborative work environment (*collaboration space*) accessible via a network using a client/server arrangement. The server stores information relating to a project or a set of projects, referred to as a facility (*place*), in a database (20), which can be object-oriented. There can be multiple servers hosting multiple facilities. Each facility is viewed by a user as a directory of eRoom (*room*) pages (Col. 3, lines 14-37).

Salas also teaches in Fig. 2 that the structure of one facility in the collaboration space is basically that of a hierarchical, object based (*object model*) file structure each entry providing at least one link to an eRoom front page (24), which in turn may contain other eRoom pages (27), folders containing database objects (28) or files (29) (Col. 3, lines 33-37). It would have been obvious to one of ordinary skill in the art at the time of invention to combine the teachings of Cannata and Salas, as both inventions relate to the creation of collaborative web-based environments through templates. Adding the teaching of Salas allows for creating/updating collaborative web-based workspaces without the need to be directly connected to the server.

**In regard to dependent Claim 6, Cannata fails to teach *said room object visible as a unique identifier in uniform resource locators (urls)*. However, Salas teaches that users interact with eRooms by using Web browsers in a traditional manner. That is, users may traverse a hyperlink to access an eRoom, or users may directly enter a URL address into the browser (Col. 6, lines 40-43; Fig. 4). Salas implies that one can use a URL to completely specify a document. It would have been obvious to one of ordinary skill in the art at the time of invention to combine the teachings of Cannata and Salas,**



as both inventions relate to the creation of collaborative web-based environments through templates. Adding the teaching of Salas allows for creating/updating collaborative web-based workspaces without the need to be directly connected to the server.

**In regard to independent Claim 11 (and similarly independent Claims 17-18),** Claim 11 (and similarly Claims 17, and 18) reflects the method for integrating forms including fields defining the schema of a collaboration space into said collaboration space, as claimed in Claim 1, and is rejected along the same rationale.

In addition, Cannata fails to explicitly teach *creating a form in html separate from said place*. However, Salas teaches that a file (can be an html file, html form, word processing file, etc...) is edited by first selecting the file from the given eRoom, at which point it gets downloaded to the client, or it can already exist on the client's mass storage (hard disk, tape drive, memory), or it could be a newly created file (Col. 12, lines 7-22, 47-67; Col. 13, lines 1-26). Once the file has been downloaded, or if the file was already present in local mass storage, the Watcher launches the application used to edit the file (step 604). The indicated application may be determined using the Object Linking Embedding standard (OLE), the file suffix (three characters in a DOS-based system), or the server 14 may store file metadata which is transmitted together with the file and indicates which application should be used to open and edit the file. If the server 14 stores file metadata, a list of alternate applications may be stored, either on the server 14 or the client workstation 12, so that if a client workstation does not have access to a

first application, other applications are specified which may be used (Col. 12, lines 47-59).

**In regard to independent Claim 13**, Claim 13 reflects the method for integrating forms including fields defining the schema of a collaboration space into said collaboration space, as claimed in Claim 1, and is rejected along the same rationale.

In addition, neither Cannata nor Salas fails to teach that *at said place, replacing head and title tags in said document with place tags*. However, it would have been obvious to one of ordinary skill in the art at the time of invention to modify the file once uploaded to a collaboration space simply to make the file usable in a collaborative environment.

7. Claims 7-9 are rejected under 35 U.S.C. 103(a) as being unpatentable over Cannata in view of Salas, and in further view of Hanson et al. (hereinafter Hanson, U.S. Patent No. 5,956,736).

**In regard to dependent Claim 7**, neither Cannata nor Salas explicitly teach *using said field object to construct html formatted input fields in forms*. However, Hanson teaches building an HTML document using an object oriented editor and HTML objects (Abstract).

Hanson also teaches that having selected an HTML object from either the user or widgets panel of the HTML Palette (500) for inclusion in the collection of objects comprising the current Web document in the page structure panel, a user can edit or modify the properties associated with the object via a context sensitive object editor

window. For example, with reference to Fig. 6C, a user clicks on the representation of the HTML object HTMLText (622) in the page structure panel (608) of Viewer window (600). The user then drags the representation of the HTML object HTMLText (622) and drops it into Object Editor window (610). The object display panel shows an icon and an object name HTMLText (623) representing the HTML object. Furthermore, the properties display panel (614) provides access to the font styles (625), information styles (626) and text string (624) properties associated with the HTML object HTMLText. The user can manipulate any of these properties, as is illustrated by creating the text string "This is a header" in the property display panel (Col. 12, lines 1-18).

Hanson also teaches that the present invention allows the Web page created by the object-oriented HTML based editor to be saved as an object rather than a data file in a file system. The object can then be published by a server on the Web as is, or the object can output its contents in a format requested by a client on the Web, e.g., an HTML formatted document (Col. 4, lines 40-45). It would have been obvious to one of ordinary skill in the art at the time of invention to combine the teachings of Cannata, Salas and Hanson as all three inventions relate to the creation of html documents. Hanson adds the benefit of further defining each component of a web page (such as form input fields) in the form of objects whose attributes can be manipulated.

**In regard to dependent Claim 8, neither Cannata nor Salas specifically teach *defining said form object using said form including imported hyperlinks*. However,**

Hanson teaches building an HTML document using an object oriented editor and HTML objects (Abstract).

Hanson also teaches that having selected an HTML object from either the user or widgets panel of the HTML Palette (500) for inclusion in the collection of objects comprising the current Web document in the page structure panel, a user can edit or modify the properties associated with the object via a context sensitive object editor window. For example, with reference to Fig. 6C, a user clicks on the representation of the HTML object HTMLText (622) in the page structure panel (608) of Viewer window (600). The user then drags the representation of the HTML object HTMLText (622) and drops it into Object Editor window (610). The object display panel shows an icon and an object name HTMLText (623) representing the HTML object. Furthermore, the properties display panel (614) provides access to the font styles (625), information styles (626) and text string (624) properties associated with the HTML object HTMLText. The user can manipulate any of these properties, as is illustrated by creating the text string "This is a header" in the property display panel (Col. 12, lines 1-18).

Hanson also teaches that the present invention allows the Web page created by the object-oriented HTML based editor to be saved as an object rather than a data file in a file system. The object can then be published by a server on the Web as is, or the object can output its contents in a format requested by a client on the Web, e.g., an HTML formatted document (Col. 4, lines 40-45). It would have been obvious to one of ordinary skill in the art at the time of invention to combine the teachings of Cannata,

Salas and Hanson as all three inventions relate to the creation of html documents.

Hanson adds the benefit of further defining each component of a web page (such as form input fields) in the form of objects whose attributes can be manipulated.

**In regard to dependent Claim 9**, neither Cannata nor Salas explicitly teach *parsing said form, finding linked files and processing uniform resource locators to generate a form object; saving to a page object said form, said linked files, and said form object; and displaying at said browser said page with said form object in read mode and said form in edit mode*. However, Hanson teaches an object comprising a web document being executed by a server, having been previously uploaded to the server from an object-oriented web editor client, and subsequently requested by a client for display (Col. 13, lines 48-57). So, the object (web page) was created by an editor and uploaded to the server as an object. The web page (object) consists of other object (each HTML tag, for example). So, the web object (form) is executed (involving parsing) and tags are converted to whatever formats the client requests (e.g., client can request an HTML page, or other characteristics of the object) via handlers. It would have been obvious to one of ordinary skill in the art at the time of invention to combine the teachings of Cannata, Salas and Hanson as all three inventions relate to the creation of html documents. Hanson adds the benefit of further defining each component of a web page (such as form input fields) in the form of objects whose attributes can be manipulated.

8. Claims 10-12, and 17-18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Cannata in view of Salas, and in further view of Kagle (U.S. Patent No. 6,779,153).

**In regard to dependent Claim 10, Cannata fails to teach *downloading said images into a local directory in the same folder as said html page or skin*. However, Salas teaches that file download and subsequent upload, if necessary, is managed by a background daemon. Alternatively, file upload and download may be managed by a separately executing program; the only requirement is that the file upload/download application executes separately from the browser application, so that premature exiting of the browser program is handled appropriately by upload/download code (Col. 12, lines 15-22). It would have been obvious to one of ordinary skill in the art at the time of invention to combine the teachings of Cannata and Salas as both inventions relate to collaborative work environments. Adding the teaching of Salas allows for uploading/downloading server workspaces to maintain local versions.**

**Cannata and Salas fail to teach *uploading said linked and download images from said local directory automatically when said page or skin is uploaded*. However, Kagle teaches that pictorial/image information can be entered as a pointer to a locally stored image, which is uploaded with the completed page layout (Col. 6, lines 53-60). Here, a PDA uploads a page layout and an image referred to in the page layout. It would have been obvious to one of ordinary skill in the art at the time of invention to combine the teachings of Cannata, Salas, and Kagle as all three inventions relate to the creation of**

web pages. The benefit of Kagle's teaching is to allow for more efficient uploading by uploading all components of a web site in the same upload session.

**In regard to dependent Claim 12**, Claim 12 contains subject matter similar to that found in Claim 10, and is rejected along similar lines of reasoning.

In addition, Cannata fails to teach *importing an original html file, which contains an image tag, and the related image files into collaboration space*. However, Salas teaches that file uploading may be done in the background or in the foreground. If done in the foreground the user will be blocked from further work on that file until the upload/download is complete. Once the upload is complete, the server 14 updates metadata stored in its database 20 that is associated with the file, for example, any edit lock set by the editing user is released (Col. 132, lines 60-67). Also, once the user is finished editing the file, it may be uploaded to the server 14 to allow other users access to it. The user signals that the file should be transmitted to the server 14 by *dragging the file onto an eRoom* displayed by the browser (step 704). *Dropping the file into the displayed eRoom* invokes an ActiveX control or a background daemon process, which manages the upload of the file to the server 14 (Col. 13, lines 19-26). It would have been obvious to one of ordinary skill in the art at the time of invention to combine the teachings of Cannata and Salas as both inventions relate to collaborative work environments. Adding the teaching of Salas allows for creating/updating server workspaces without interfering with other collaborators.

Salas fails to teach *parsing said original html file to find linked images; creating a modified html file by modifying said image tag in said html file to refer to the copy of the*

*image file attached to document on the server; saving said original html file and said modified html file on a collaboration space page object.* However, Kagle teaches in Fig. 8 a process whereby template files are created on a client computer (a PDA in this case). Template files and any images, audio, etc. that are referenced in the template files are uploaded to a server. The server processes the templates and builds an html page (see Fig. 8). It would have been obvious to one of ordinary skill in the art at the time of invention to combine the teachings of Cannata, Salas, and Kagle as all three inventions relate to the creation of web pages. The benefit of Kagle's teaching is to allow for more efficient uploading by uploading all components of a web site in the same upload session.

Kagle does not discuss *displaying said page object with said original html file in edit mode and said modified html file in read mode.* However, it would have been obvious to one of ordinary skill in the art at the time of invention to leave the created html page in a read-only mode because it was created using a template. The whole reason for having a template is that it allows one to easily incorporate changes to the web page without having to know the details of html. Likewise, leaving the template file in an edit mode allows for those changes, when they occur to be made in order to regenerate the html file with the changes.



### ***Response to Arguments***

9. Applicant's arguments with respect to claims 1-6, 13-16, and 19 have been considered but are moot in view of the new ground(s) of rejection.

Respectfully, the Applicant's main argument is that the prior art of Salas fails to utilize HTML forms to define the data model of a collaboration space by automating the creation of fields within the collaboration space. The Examiner agrees to the extent that Salas is using templates (which could be interpreted as a form) to describe the contents of web page(s) that are then uploaded to a web server and processed to create HTML pages. However, the creation of web pages by such a method can be interpreted as the creation of a collaboration space in the sense that multiple developers could add their own content to the site and collaboratively build the site. Nevertheless, the Examiner now introduces the prior art of Cannata et al. which, when combined with Salas, better teaches the claimed limitation

### ***Conclusion***

10. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not

mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

11. Any inquiry concerning this communication or earlier communications from the examiner should be directed to James H. Blackwell whose telephone number is 571-272-4089. The examiner can normally be reached on Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Heather R. Herndon can be reached on 571-272-4136. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

12. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

James H. Blackwell

*William L. Bashore*  
**WILLIAM BASHORE**  
**PRIMARY EXAMINER**  
8/5/2005